Constructing and Sampling Directed Graphs with Linearly Rescaled Degree Matrices

Yunxinag Yan
yyan6@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

Meng Jiang
mjiang2@nd.edu
University of Notre Dame
Notre Dame, Indiana, USA

ABSTRACT

In recent years, many large directed networks such as online social networks are collected with the help of powerful data engineering and data storage techniques. Analyses of such networks attract significant attention from both the academics and industries. However, analyses of large directed networks are often time-consuming and expensive because the complexities of a lot of graph algorithms are often polynomial with the size of the graph. Hence, sampling algorithms that can generate graphs preserving properties of original graph are of great importance because they can speed up the analysis process. We propose a promising framework to sample directed graphs: Construct a sample graph with linearly rescaled Joint Degree Matrix (JDM) and Degree Correlation Matrix (DCM). Previous work shows that graphs with the same JDM and DCM will have a range of very similar graph properties. We also conduct experiments on real-world datasets to show that the numbers of non-zero entries in JDM and DCM are quite small compared to the number of edges and nodes. Adopting this framework, we propose a novel graph sampling algorithm that can provably preserves in-degree and out-degree distributions, which are two most fundamental properties of a graph. We also prove the upper bound for deviations in the joint degree distribution and degree correlation distribution, which correspond to JDM and DCM. Besides, we prove that the deviations in these distributions are negatively correlated with the sparsity of the JDM and DCM. Considering that these two matrices are always quite sparse, we believe that proposed algorithm will have a better-than-theory performance on real-world large directed networks.

ACM Reference Format:
Yunxinag Yan and Meng Jiang. 2022. Constructing and Sampling Directed Graphs with Linearly Rescaled Degree Matrices. In *Proceedings of KDD Undergraduate Consortium (KDD-UC).* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Analysis of large directed networks has always been an important topic in data science applications. Such analyses range from the spread of COVID-19 [11], twitter fake account detection [7], to automobile insurance fraud detection [30]. However, as researchers show more and more interest in the emerging large directed networks, their attempts to conduct analysis on the whole graph are always hindered by unbearably long running time and running cost because the time complexities of a lot of graph algorithms are polynomial to the size of the graph. One solution is to create a representative sample graph out of the large graph while preserving important properties of the original graph such as degree distributions, degree correlations, and clustering coefficients [20].

A number of graph sampling algorithms have been proposed in the past. They can be classified into several categories [16, 27], such as node sampling [1, 16, 29], edge sampling [2, 13], and exploration-based sampling [5, 8–10, 15–19, 22, 24–26, 28, 32, 33]. However, most of them depend on heuristics and thus can not guarantee the performance (i.e., how the important properties are preserved). There is a previous generation-based algorithm [12] which can preserve in/out-degree distributions but its complexity is $O(N * E)$ (N: number of nodes, E: number of edges), making it impractical to large directed networks.

Present Work. We propose a new framework of sampling large directed graphs: Constructing a sample graph using linearly rescaled JDM and DCM (Section 3.2). We show that the numbers of non-zero entries in JDM and DCM calculated from large directed network are always way smaller than the number of edges or nodes (Section 4). This property gives algorithms that adopt this framework great potential to be **efficient** because the constructing process only involves in iterating each non-zero entry of JDM and DCM. Additionally, previous work [31] shows that graphs with the same JDM will have the **same in-degree and out-degree distributions** as the original graph and may share a lot of **similar properties** of the original graph, for example, Dyad Census, Triad Census, Shortest Path Distribution, Eigenvalues, Average Neighbor Degrees, etc. Adopting this framework, we propose a sampling algorithm that builds a simple directed graph with given JDM using the D2K construction method showed in [31] (Section 3.2). We prove that this algorithm can preserve degree distributions (Section 5) and the deviation of distributions from the original graph has an upper bound which is negatively correlated to the sparsity of the joint degree matrix (Section 6).

2 RELATED WORK

There are two lines of the past work that are related to our work.

2.1 Construction of Directed 2K Graphs (D2K)

The taxonomy of graph construction tasks: dK-series[6, 21, 23] provides an elegant way to trade off accuracy (in terms of graph properties) for complexity (in generating graph realizations). The constructions of both directed and undirected dK-graphs are well

KDD-UC, August 14-18, 2022, Washington DC, USA

ACM ISBN 978-1-4503-9999-9/18/06…$15.00

understood (i.e., efficient algorithms and realizability conditions are known) for 0K (graphs with prescribed number of vertices and edges), 1K (graphs with a given degree distribution) and 2K (graphs with a given joint degree matrix). In this taxonomy system, d refers to the dimension of the distribution. Specifically, the D2K algorithm we use in our sampling algorithm constructs a graph $\mathcal{G}$ with a target JDM $\mathbf{A}^{\odot}$ and a target DCM $\mathbf{B}^{\odot}$.

2.2 Construction of Matrices with Line Sums

Matrices with prescribed row and column sum vectors have always been an object of interest for mathematicians in the field of matrix theory. A lot of work has been done including the construction of these matrices and the necessary and sufficient conditions for such construction [3, 4]. We use the graphical conditions and construction algorithm from [4] as our building block of the sampling algorithm. The matrix construction algorithm takes row and column sum vectors and the integer upper bound of each entry value, p as inputs and gives a matrix that satisfies these constraints as the output.

3 PROPOSED METHOD

3.1 Matrices and Vectors

We first introduce the definition of Joint Degree Matrix (JDM) and Degree Correlation Matrix (DCM). Joint Degree Matrix, A is defined as: a matrix where each entry is the number of **edges** that have the respective out-degree and in-degree pattern. Degree Correlation Matrix B is defined as: a matrix where each entry is the number of **nodes** that have the respective out-degree and in-degree:

$$A = \{a_{ij} | a_{ij} = |\{(v_1, v_2) \in E | d^{out}_{v_1} = i, d^{in}_{v_2} = j\}|\}, \quad (1)$$

$$B = \{b_{ij} | b_{ij} = |\{v \in V | d^{out}_{v} = i, d^{in}_{v} = j\}|\}. \quad (2)$$

A and B can be obtained by first looping through all the edges and calculating the in/out-degrees of each node. Then, we can get A and B by counting the nodes and edges that have the patterns described above with another iteration.

Suppose A is a matrix of size m by n, r_i and s_j are the sum of row i and column j of A: $r_i = \sum_{j=1}^{n} a_{ij}$ and $s_j = \sum_{i=1}^{m} a_{ij}$, where $i = 1, \cdots m$, $j = 1, \cdots, n$. Then $\sigma_{\mathcal{R}}(\cdot)$ and $\sigma_{C}(\cdot)$ are the row and column sum functions:

$$\sigma_{\mathcal{R}}(A) = (r_1, \cdots, r_m), \quad (3)$$

$$\sigma_{C}(A) = (s_1, \cdots, s_n). \quad (4)$$

3.2 Sampling Framework

We propose a new graph sampling framework: Construction based Sampling with Joint Degree Matrix (JDM) and Degree Correlation Matrix (DCM). In general, this framework utilizes the favorable property: preserving JDM and DCM can not only guarantee that in/out-degree distributions will be preserved but also help capture the degree pairing patterns in edges and nodes and hence capturing more fundamental graph properties. The process of the framework is illustrated in **Figure 1** below. For a given graph, we first calculate JDM and DCM and then we conduct the sampling process: we multiply each entry of the matrices by a sample coefficient k and intergerize it with functions such as $floor()$, $ceiling()$ or $round()$. After that, we use a certain construction method that can build a graph that approximately satisfies the linearly rescaled JDM and DCM. Note that the choice of construction method affects the overall performance of the algorithm a lot and in different scenarios, different construction methods may be more preferable because of the existence of accuracy-efficiency tradeoff.

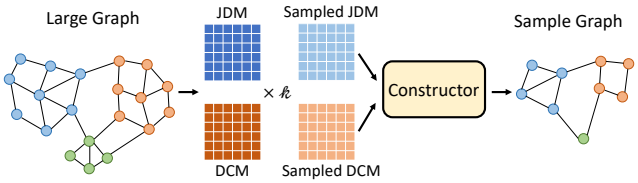

Figure 1: The process of the Construction based Sampling with JDM and DCM Framework

3.3 Sampling Algorithm

In this section we introduce one sampling algorithm that adopts the framework above. This algorithm uses *D2K* as its construction method, therefore it also includes the process of adjusting JDM to satisfy the condition of *D2K*. The sampling algorithm takes Joint Degree Matrix A, Degree Correlation Matrix B and Sample Coefficient k as inputs and constructs a simple direct graph preserving in/out-degree distributions, joint degree distributions and degree correlation distribution (latter two are defined in Section 5.1). It proceeds by first rescales both A and B using k. Then by arithmetic operations it gets the row and column sum vectors $\mathbf{r}_{\delta}$ and $\mathbf{c}_{\delta}$ as well as entry upper bound p of the Adjustment Matrix D. After that, we use *GRAPHICAL* condition to decide if $\mathbf{r}_{\delta}$, $\mathbf{c}_{\delta}$ and p are realizable. If *GRAPHICAL* returns **TRUE** we use *CONSTRUCT* to build such D and add it to linearly-rescaled Joint Degree Matrix A. Finally, we use *D2K* algorithm to build a simple, directed sample graph $\mathcal{G}'$ with target Joint Degree Matrix (after modification) $\mathrm{A}^{\odot}$ and target Degree Correlation Distribution Matrix $\mathrm{B}^{\odot}$. Note that detailed description of *GRAPHICAL*, *CONSTRUCT* and *D2K* can be found in [4] and [31].

4 SPARSITY OF JDM AND DCM

In the proposed sampling algorithm, we use two matrices frequently: Joint Degree Matrix A and Degree Correlation Matrix B. As mentioned above, these two matrices are always quite sparse. We illustrate this property by conducting experiments on a variety of different real-world networks that belong to different categories and have different sizes. All datasets are from a free public project called **The KONECT Project** [14]. The experimental data are demonstrated in **Table 1**.

Experimental datasets include five online social networks and three citation networks with node sizes ranging from 10^3 to 10^7 and edge sizes ranging from 10^4 to 10^7. The meaning of each column is, N: the number of nodes, E: the number of edges, # DCM: the number non-zero entris in Degree Correlation Matrix, # JDM: the number of non-zero entris in Joint Degree Matrix, % DCM: $\frac{\#DCM}{N}$, % JDM: $\frac{\#JDM}{E}$.

From the data we can see that % DCM and % JDM are below 30% for all datasets, and as the size of graph grows bigger, these

Algorithm 1: Sampling Algorithm

input : Joint Degree Matrix A, Degree Correlation Matrix B, Sample Coefficient k

output : A simple directed sample graph $\mathcal{G}'$ preserving four distributions

$A' \leftarrow \{a'_{ij} | a'_{ij} = \lfloor \frac{1}{k} a_{ij} \rfloor, \forall a_{ij} \in \mathrm{A}\}$
$B' \leftarrow \{b'_{ij} | b'_{ij} = \lceil \frac{1}{k} b_{ij} \rceil, \forall b_{ij} \in \mathrm{B}\}$
$\mathbf{r}_{A'} \leftarrow \sigma_{\mathcal{R}}(\mathrm{A}')$, $\mathbf{r}_{B'} \leftarrow \sigma_{\mathcal{R}}(\mathrm{B}')$, $\mathbf{c}_{A'} \leftarrow \sigma_C(\mathrm{A}')$, $\mathbf{c}_{B'} \leftarrow \sigma_C(\mathrm{B}')$
for $i = 1, \cdots, m$ **do**
  $\widetilde{\mathbf{r}_{B'}}^{(i)} := \mathbf{r}_{B'}^{(i)} \cdot i$
end
for $j = 1, \cdots, n$ **do**
  $\widetilde{\mathbf{c}_{B'}}^{(j)} := \mathbf{c}_{B'}^{(j)} \cdot j$
end
$\mathbf{r}_\delta \leftarrow \widetilde{\mathbf{r}_{B'}} - \mathbf{r}_{A'}$, $\mathbf{c}_\delta \leftarrow \widetilde{\mathbf{c}_{B'}} - \mathbf{c}_{A'}$
$\mathrm{L} \leftarrow \{l_{ij} | l_{ij} = \mathrm{r}_{B'}^{(i)} \cdot \mathrm{c}_{B'}^{(j)} - b'_{ij}, \forall b'_{ij} \in B'\}$
$p := \mathbf{min}_{l_{ij} \in L}\{l_{ij} - a'_{ij}\}$
if $GRAPHICAL(\mathbf{r}_\delta, \mathbf{c}_\delta, p)$ == **TRUE then**
  $D \leftarrow CONSTRUCT(\mathbf{r}_\delta, \mathbf{c}_\delta, p)$
else
  return FALSE
end
$\mathrm{A}^{\odot} \leftarrow \mathrm{A}' + \mathrm{D}$, $\mathrm{B}^{\odot} \leftarrow \mathrm{B}'$
$\mathcal{G}'(\mathcal{V}', \mathcal{E}') \leftarrow D2K(\mathrm{A}^{\odot}, \mathrm{B}^{\odot})$
return $\mathcal{G}'$

percentages tend to drop. For datasets that have a node size larger than 10^6, the percentages are around 1%. This observation shows that JDM and DCM are usually quite sparse and the number of non-zero entries in them are quite small compared to the size of nodes and edges, especially when the graph size exceeds certain threshold.

5 PROOF OF VALIDITY

5.1 Notations

First, we introduce some notations that will be used in the following sections. $\mathcal{V} = \{v_i\}$ is the set of vertices. $\mathcal{E} \subseteq \{(v_1, v_2) | (v_1, v_2) \in \mathcal{V}^2, v_1 \neq v_2\}$ denotes the directed edge set where each element belongs to the Cartesian square of set $\mathcal{V}$. (v_1, v_2) represents an edge that is pointing from v_1 to v_2. $|\mathcal{V}|$ and $|\mathcal{E}|$ are number of vertices and edges in the graph. We define $V_{k,p} = \{v \in \mathcal{V} | d_v^p = k\} \subset \mathcal{V}$, $p \in \{in, out\}$ as a vertices subset of $\mathcal{V}$, where all vertices' in-degree (or out-degree, depends on the value of p) equals to k. For example, $V_{1,in}$ denotes the subset of nodes with in-degree 1. Note that in order to avoid confusion we doesn't use N and E to represent the vertices and edges number in section 5 and 6.

We denote $P(k, p) = \frac{|V_{k,p}|}{|\mathcal{V}|}$ where $k = 0, 1, 2, \cdots; p \in \{in, out\}$ as the degree distribution, which equals to the fraction of nodes with certain in-degree (or out-degree). So, $P(k, in)$ is the **in-degree distribution** and $P(k, out)$ is the **out-degree distribution**. For example, $P(1, in) = \frac{|V_{1,in}|}{|\mathcal{V}|}$ denotes for the fraction of nodes with in-degree 1.

Furthermore, we define $P(i, j) = \frac{|\{v \in \mathcal{V} | d_v^{out} = i, d_v^{in} = j\}|}{|\mathcal{V}|}$, $i, j = 0, 1, 2, \cdots$ as the **degree correlation distribution**, which equals to the fraction of nodes having certain out-degree and in-degree. Given a pair of nodes v_i and v_j that are connected, we denotes $\tilde{P}(i, j) = \frac{|\{(v_1, v_2) \in \mathcal{E} | d_{v_1}^{out} = i, d_{v_2}^{in} = j\}|}{|\mathcal{E}|}$ as their **joint degree distribution**.

5.2 Consistency

In this section we will prove the following important fact:

- If the *GRAPHICAL* checking gives us **TRUE**, the condition of *D2K* will be automatically satisfied.

In [31], Balint Tillman et al. give the condition for target JDM: $\mathrm{A}^{\odot}$ and target DCM: $\mathrm{B}^{\odot}$ to be realizable(i.e. graphical). We give the equivalent D2K CONDITION, which is a modified version from their original condition to match the changes of definition and notation in this paper.

Theorem 5.1 (D2K CONDITION). *Let $\mathrm{A}^{\odot}$ be the joint degree matrix, $\mathrm{B}^{\odot}$ be the degree correlation matrix both of size m by n. There is a graph $\mathcal{G}$ satisfying $\mathrm{B} = \mathrm{B}^{\odot}$ and $\mathrm{A} = \mathrm{A}^{\odot}$ if and only if $\forall i = 1, \cdots, m; j = 1, \cdots, n$,*

$$|V_{i,out}| = \sum_j \frac{a_{ij}^{\odot}}{i} = \sum_j b_{ij}^{\odot}, \tag{5}$$

$$|V_{j,in}| = \sum_i \frac{a_{ij}^{\odot}}{j} = \sum_i b_{ij}^{\odot}. \tag{6}$$

$$a_{ij}^{\odot} + b_{ij}^{\odot} \leq |V_{i,out}| \cdot |V_{j,in}|, \tag{7}$$

We show that the D2K CONDITION will be automatically satisfied by the process of our sampling algorithm if *GRAPHICAL* gives **TRUE**.

Next, we divide the task into two lemmas and provide proofs for each of them.

Lemma 5.2 (Condition 1 and 2). *If $GRAPHICAL(\mathbf{r}_\delta, \mathbf{c}_\delta, p)$ == **TRUE**, by following the steps of **Algorithm 1**, the first and second D2K CONDITION will be satisfied.*

Proof. Equation (5) and (6) are equivalent to the following identity:

$$\sum_j a_{ij}^{\odot} = i \cdot \sum_j b_{ij}^{\odot} \tag{8}$$

$$\sum_i a_{ij}^{\odot} = j \cdot \sum_i b_{ij}^{\odot} \tag{9}$$

Without loss of generality, we only need to show (8) holds for an arbitrary choice of i and j. Note that we have:

$$a_{ij}^{\odot} = a'_{ij} + d_{ij}, b_{ij}^{\odot} = b'_{ij}$$

$$\therefore (8) \longleftrightarrow \sum_j a'_{ij} + \sum_j d_{ij} = i \cdot \sum_j b'_{ij} \tag{10}$$

According to **Algorithm 1**, (10) is equivalent to:

$$\mathbf{r}_{A'}^{(i)} + \mathbf{r}_{\delta}^{(i)} = i \cdot \mathbf{r}_{B'}^{(i)} \tag{11}$$

Category	Name	N	E	# DCM	# JDM	% DCM	% JDM
Online Social Networks	FilmTrust trust	874	1853	102	413	11.67%	22.29%
	CiaoDVD trust	4658	40133	834	11127	17.90%	27.73%
	Epinions	75879	508837	4398	96382	5.80%	18.94%
	Twitter (ICWSM)	465017	834797	1172	25501	0.25%	3.05%
	Youtube links	1138499	4942297	8859	322950	0.78%	6.53%
Citation Networks	DBLP	12590	49759	769	5201	6.11%	10.45%
	arXiv hep-ph	34546	421578	3925	28106	11.36%	6.67%
	CiteSeer	384413	1751463	4030	27399	1.05%	1.56%

Table 1: The Number of non-negative entries in JDM and DCM compared to the number of edges and nodes in real-world networks

$$\because \mathbf{r}_{\delta}^{(i)} = \widetilde{\mathbf{r}_{B'}^{(i)}} - \mathbf{r}_{A'}^{(i)} = i \cdot \mathbf{r}_{B'}^{(i)} - \mathbf{r}_{A'}^{(i)}$$
$$\therefore \mathbf{r}_{A'}^{(i)} + \mathbf{r}_{\delta}^{(i)} = \mathbf{r}_{A'}^{(i)} + i \cdot \mathbf{r}_{B'}^{(i)} - \mathbf{r}_{A'}^{(i)} = i \cdot \mathbf{r}_{B'}^{(i)} \quad (12)$$

This completes the proof of LEMMA 5.2

□

Lemma 5.3 (Condition 3). *If GRAPHICAL*$(\mathbf{r}_{\delta}, \mathbf{c}_{\delta}, p)$ == **TRUE**, *following the steps of* ***Algorithm 1****, the third D2K CONDITION will be satisfied.*

Proof. From the proof of LEMMA 5.2, we know that:

$$\because |V_{i,out}| \cdot |V_{j,in}| = ij \cdot \sum_j b_{ij}^{\odot} \cdot \sum_i b_{ij}^{\odot} = ij \cdot \sum_j b'_{ij} \cdot \sum_i b'_{ij}$$

and

$$a_{ij}^{\odot} + b_{ij}^{\odot} = a'_{ij} + d_{ij} + b'_{ij}$$

Thus, (7) is equivalent to:

$$d_{ij} \le ij \cdot \sum_j b'_{ij} \cdot \sum_i b'_{ij} - b'_{ij} - a'_{ij} = l_{ij} - a'_{ij} \quad (13)$$

According to algorithm *CONSTRUCT*, we have

$$d_{ij} \le p = \mathbf{min}_{l_{ij} \in L}\{l_{ij} - a'_{ij}\}$$
$$\therefore d_{ij} \le l_{ij} - a'_{ij}, \forall l_{ij} \in L$$

This completes the proof for LEMMA 5.3

□

5.3 Preserving Distributions

In this section we show that the sample graph has the **same in/out-degree distribution** and the **same degree correlation distribution** as the original graph $\mathcal{G}$, i.e. $P(k, in)$, $P(k, out)$ and $P(i, j)$. Additionally, we will also show that the joint degree distribution $\tilde{P}'(i, j)$ of sample graph will also be similar to the joint degree distribution $\tilde{P}(i, j)$ of the original graph with an upper bound that will be studied in Section 6.

We construct two important variable $\mathring{a_{ij}}$ and $\mathring{b_{ij}}$ as below:

$$\mathring{a_{ij}} = \frac{1}{k} a_{ij}, \mathring{b_{ij}} = \frac{1}{k} b_{ij}. \quad (14)$$

By definition of a'_{ij} and b'_{ij} we have:

$$a'_{ij} = \lfloor \frac{1}{k} a_{ij} \rfloor = \lfloor \mathring{a_{ij}} \rfloor, b'_{ij} = \lceil \frac{1}{k} b_{ij} \rceil = \lceil \mathring{b_{ij}} \rceil \quad (15)$$

We get two important inequalities:

$$\mathring{a_{ij}} - 1 < a'_{ij} \le \mathring{a_{ij}}, \mathring{b_{ij}} \le b'_{ij} < \mathring{b_{ij}} + 1 \quad (16)$$

Firstly, we show that the degree correlation distribution is preserved. From the definition of $P(i, j)$ and Degree Correlation Matrix (DCM) B we have:

$$P(i, j) = \frac{|\{v \in V | d_v^{out} = i, d_v^{in} = j\}|}{|V|} = \frac{b_{ij}}{\sum_{i,j} b_{ij}} = \frac{\frac{1}{k} b_{ij}}{\frac{1}{k} \sum_{i,j} b_{ij}} \quad (17)$$

From step 18 in **Algorithm 1** we have:

$$a_{ij}^{\odot} = a'_{ij} + d_{ij}, b_{ij}^{\odot} = b'_{ij} \quad (18)$$

By (8), (9), (14) and (15):

$$P(i, j) = \frac{\frac{1}{k} b_{ij}}{\frac{1}{k} \sum_{i,j} b_{ij}} = \frac{\mathring{b_{ij}}}{\sum_{i,j} \mathring{b_{ij}}} \approx \frac{b'_{ij}}{\sum_{i,j} b'_{ij}} = \frac{b_{ij}^{\odot}}{\sum_{i,j} b_{ij}^{\odot}} = P^{\odot}(i, j)$$

Thus the sample graph $\mathcal{G}'$ has the same degree correlation distribution and the only deviation is from the Integerization process. This deviation is quantified in Section 6.

Note that $P(i, j)$ is the joint probability mass function of $P(k, in)$ and $P(k, out)$:

$$P(k, in) = P(\cdot, k) = \sum_i P(i, k); P(k, out) = P(k, \cdot) = \sum_j P(k, j).$$

Therefore the sample graph $\mathcal{G}'$ will automatically also preserve the in/out-degree distribution of the original graph when the degree correlation distribution is preserved.

From the definitions of joint degree distribution and Joint Degree Matrix (JDM), we have the following observation:

$$\tilde{P}(i, j) = \frac{|\{(v_1, v_2) \in E | d_{v_1}^{out} = i, d_{v_2}^{in} = j\}|}{|E|} = \frac{a_{ij}}{\sum_{i,j} a_{ij}}. \quad (19)$$

Because of (8), (14) and (15) , similarly, we have

$$\tilde{P}(i, j) = \frac{a_{ij}}{\sum_{i,j} a_{ij}} = \frac{\frac{1}{k} a_{ij}}{\sum_{i,j} \frac{1}{k} a_{ij}} = \frac{\mathring{a_{ij}}}{\sum_{i,j} \mathring{a_{ij}}} \approx \frac{a'_{ij}}{\sum_{i,j} a'_{ij}}.$$

Because $a_{ij}^{\odot} = a'_{ij} + d_{ij}$, the joint degree distribution of sample graph $\tilde{P}^{\odot}(i, j) = \frac{a_{ij}^{\odot}}{\sum_{i,j} a_{ij}^{\odot}}$ is similar but different from the joint degree distribution of the original graph $\frac{a'_{ij}}{\sum_{i,j} a'_{ij}}$. In next section, we will show that the deviation caused by d_{ij} also has an upper bound.

6 DEVIATION ANALYSIS

In this section, we attempt to quantify the deviations of degree distributions by giving an upper bound for them.

6.1 Integerization Deviation

From (16), we can derive the following inequalities:

$$[\textbf{Deviation}]\frac{\mathring{b_{ij}}}{\sum_{i,j}\mathring{b_{ij}}+mn-1} < \frac{b'_{ij}}{\sum_{i,j}b'_{ij}} < \frac{\mathring{b_{ij}}+1}{\sum_{i,j}\mathring{b_{ij}}+1} \quad (20)$$

and

$$[\textbf{Deviation}]\frac{\mathring{a_{ij}}-1}{\sum_{i,j}\mathring{a_{ij}}-1} < \frac{a'_{ij}}{\sum_{i,j}a'_{ij}} < \frac{\mathring{a_{ij}}}{\sum_{i,j}\mathring{a_{ij}}-mn+1} \quad (21)$$

where m is the number of rows and n is the number of columns. From **Algorithm 1** we know that $b^{\odot}_{ij} = b'_{ij}$. Thus (20) quantifies the deviation brought by Integerization for degree correlation distribution $P^{\odot}(i,j)$.

$$\frac{\mathring{b_{ij}}}{\sum_{i,j}\mathring{b_{ij}}+mn-1} < \frac{b^{\odot}_{ij}}{\sum_{i,j}b^{\odot}_{ij}} = P^{\odot}(i,j) < \frac{\mathring{b_{ij}}+1}{\sum_{i,j}\mathring{b_{ij}}+1} \quad (22)$$

From **Algorithm 1** we know that $a^{\odot}_{ij} = a'_{ij} + d_{ij}$. Therefore, for joint degree distribution $\tilde{P}^{\odot}(i,j)$, however, we still need to consider the change in value brought by d_{ij}. We know from algorithm CONSTRUCT that $0 \le d_{ij} \le p$

$$\frac{a'_{ij}}{\sum_{i,j}a'_{ij}+mnp-p} < \frac{a'_{ij}+d_{ij}}{\sum_{i,j}a'_{ij}+\sum_{i,j}d_{i,j}} = \frac{a^{\odot}_{ij}}{\sum_{i,j}a^{\odot}_{ij}} < \frac{a'_{ij}+p}{\sum_{i,j}a'_{ij}+p}$$

Combining the above inequalities with (21), we have:

$$\frac{\mathring{a_{ij}}-1}{\sum_{i,j}\mathring{a_{ij}}+mnp-p-1} < \frac{a^{\odot}_{ij}}{\sum_{i,j}a^{\odot}_{ij}} = \tilde{P}^{\odot}(i,j) < \frac{\mathring{a_{ij}}+p}{\sum_{i,j}\mathring{a_{ij}}+p-mn+1} \quad (23)$$

This quantifies the deviation of $\tilde{P}^{\odot}(i,j)$ from $\tilde{P}(i,j)$.

6.2 Effects of Sparsity

Moreover, if we take the sparsity of the graph into account, we can get a more accurate bound of deviation. This is because the deviation of Intergerization does not happen to those entries that are zero in original matrices ($a'_{ij} = \mathring{a_{ij}}, \forall a_{ij} = 0, b'_{ij} = \mathring{b_{ij}}, \forall b_{ij} = 0$). Hence, the sparsity of the original JDM and DCM will directly affect the deviation from Intergerization.

We define the sparsity coefficient of row i in matrix $A_{m\times n}$ as:

$$s^A_{\mathcal{R}}(i) = \frac{I_{(j\in\{1,\cdots,n\})}a_{ij}=0}{n}.$$

Similarly the sparsity coefficient of column j in matrix $A_{m\times n}$ is:

$$s^A_{C}(j) = \frac{I_{(i\in\{1,\cdots,m\})}a_{ij}=0}{m}.$$

I is the indicator function. Note that greater the sparsity coefficient is, more sparse that line/column will be (i.e. the fraction of 0 in that line/column). Using the definition of sparsity coefficient, we can refine the qualification inequalities (20), (23) as:

$$\frac{\mathring{b_{ij}}}{\sum_{i,j}\mathring{b_{ij}}+m'_B\cdot n'_B-1} < P^{\odot}(i,j) < \frac{\mathring{b_{ij}}+1}{\sum_{i,j}\mathring{b_{ij}}+1}, \quad (24)$$

and

$$\frac{\mathring{a_{ij}}-1}{\sum_{i,j}\mathring{a_{ij}}+m'_An'_Ap-p-1} < \tilde{P}^{\odot}(i,j) < \frac{\mathring{a_{ij}}+p}{\sum_{i,j}\mathring{a_{ij}}+p-m'_An'_A+1}, \quad (25)$$

where $m'_A = m\cdot(1-s^A_C(j))$, $n'_A = n\cdot(1-s^A_{\mathcal{R}}(i))$, $m'_B = m\cdot(1-s^B_C(j))$, $n'_B = m\cdot(1-s^B_{\mathcal{R}}(i))$.

Note that greater the sparsity coefficient, smaller the deviation of both $P^{\odot}(i,j)$ from $P(i,j)$ and $\tilde{P}^{\odot}(i,j)$ from $\tilde{P}(i,j)$. This property has strong realistic meaning because we show in section 4 that JDM and DCM are always quite sparse and they tend to get more sparse as the size of the graph increases. Therefore the real performance of the sampling algorithm proposed in this paper will be better than the range presented in Section 6.1.

7 CONCLUSIONS

We propose a new sampling framework that is efficient and is able to preserve important graph properties. Based on this framework we provide a new sampling algorithm using D2K construction method. We prove that this algorithm can preserve in/out-degree distributions, joint degree distributions and degree correlation distributions. We also analyze the effects of the JDM, DCM sparsity on deviation of degree distributions and provide upper bounds that are modified with sparsity coefficients for deviations. Additionally, we use experiments to show that JDM and DCM of real-world graphs are always sparse, which lends credence to the belief that the proposed sampling algorithm will have a better-than-theory performance on real-life large directed networks. Finally, it is worth pointing out that by utilizing more efficient construction algorithms the potential of the framework may be more thoroughly realized. Hence, future work on finding faster and more accurate construction algorithms with JDM and DCM is worth conducting.

ACKNOWLEDGMENTS

This work was supported in part by NSF IIS-1849816, IIS-2142827, IIS-2146761, and ONR N00014-22-1-2507.

REFERENCES

[1] Lada A Adamic, Rajan M Lukose, Amit R Puniyani, and Bernardo A Huberman. 2001. Search in power-law networks. *Physical review E* 64, 4 (2001), 046135.
[2] Nesreen K Ahmed, Jennifer Neville, and Ramana Kompella. 2013. Network sampling: From static to streaming graphs. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 8, 2 (2013), 1–56.
[3] Richard A Brualdi. 2006. Algorithms for constructing (0, 1)-matrices with prescribed row and column sum vectors. *Discrete Mathematics* 306, 23 (2006), 3054–3062.
[4] JA Dias da Silva and Amélia Fonseca. 2009. Constructing integral matrices with given line sums. *Linear algebra and its applications* 431, 9 (2009), 1553–1563.
[5] Christian Doerr and Norbert Blenn. 2013. Metric convergence in social network sampling. In *Proceedings of the 5th ACM workshop on HotPlanet*. 45–50.
[6] Péter L Erdős, Stephen G Hartke, Leo van Iersel, and István Miklós. 2015. Graph realizations constrained by skeleton graphs. *arXiv preprint arXiv:1508.00542* (2015).
[7] Buket Erşahin, Özlem Aktaş, Deniz Kılınç, and Ceyhun Akyol. 2017. Twitter fake account detection. In *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 388–392.
[8] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2010. Walking in facebook: A case study of unbiased sampling of osns. In *2010 Proceedings IEEE Infocom*. Ieee, 1–9.
[9] Leo A Goodman. 1961. Snowball sampling. *The annals of mathematical statistics* (1961), 148–170.
[10] Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. 2008. Metropolis algorithms for representative subgraph sampling. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 283–292.
[11] Wonkwang Jo, Dukjin Chang, Myoungsoon You, and Ghi-Hoon Ghim. 2021. A social network analysis of the spread of COVID-19 in South Korea and policy implications. *Scientific Reports* 11, 1 (2021), 1–10.
[12] Hyunju Kim, Charo I Del Genio, Kevin E Bassler, and Zoltán Toroczkai. 2012. Constructing and sampling directed graphs with given degree sequences. *New Journal of Physics* 14, 2 (2012), 023012.

[13] Vaishnavi Krishnamurthy, Michalis Faloutsos, Marek Chrobak, Li Lao, J-H Cui, and Allon G Percus. 2005. Reducing large internet topologies for faster simulations. In *International Conference on Research in Networking*. Springer, 328–341.

[14] Jérôme Kunegis. 2013. Konect: the koblenz network collection. In *Proceedings of the 22nd international conference on world wide web*. 1343–1350.

[15] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2012. Beyond random walk and metropolis-hastings samplers: why you should not backtrack for unbiased graph sampling. *ACM SIGMETRICS Performance evaluation review* 40, 1 (2012), 319–330.

[16] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 631–636.

[17] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 177–187.

[18] Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. 2015. On random walk based graph sampling. In *2015 IEEE 31st international conference on data engineering*. IEEE, 927–938.

[19] Yongkun Li, Zhiyong Wu, Shuai Lin, Hong Xie, Min Lv, Yinlong Xu, and John CS Lui. 2019. Walking with perception: Efficient random walk sampling via common neighbor awareness. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 962–973.

[20] Chun Lun Lit and Inas S Khayal. 2020. Understanding Twitter telehealth communication during the COVID-19 pandemic using hetero-functional graph theory. In *2020 IEEE International Smart Cities Conference (ISC2)*. IEEE, 1–6.

[21] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. 2006. Systematic topology analysis and generation using degree correlations. *ACM SIGCOMM Computer Communication Review* 36, 4 (2006), 135–146.

[22] Arun S Maiya and Tanya Y Berger-Wolf. 2010. Sampling community structure. In *Proceedings of the 19th international conference on World wide web*. 701–710.

[23] Chiara Orsini, Marija M Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E Bassler, Zoltán Toroczkai, Marián Boguná, Guido Caldarelli, et al. 2015. Quantifying randomness in real networks. *Nature communications* 6, 1 (2015), 1–10.

[24] Alireza Rezvanian and Mohammad Reza Meybodi. 2015. Sampling social networks using shortest paths. *Physica A: Statistical Mechanics and its Applications* 424 (2015), 254–268.

[25] Bruno Ribeiro and Don Towsley. 2010. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. 390–403.

[26] Benjamin Ricaud, Nicolas Aspert, and Volodymyr Miz. 2020. Spikyball sampling: Exploring large networks via an inhomogeneous filtered diffusion. *Algorithms* 13, 11 (2020), 275.

[27] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. Little ball of fur: a python library for graph sampling. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3133–3140.

[28] Benedek Rozemberczki and Rik Sarkar. 2018. Fast sequence-based embedding with diffusion graphs. In *International Workshop on Complex Networks*. Springer, 99–107.

[29] Michael PH Stumpf, Carsten Wiuf, and Robert M May. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences* 102, 12 (2005), 4221–4224.

[30] Lovro Šubelj, Štefan Furlan, and Marko Bajec. 2011. An expert system for detecting automobile insurance fraud using social network analysis. *Expert Systems with Applications* 38, 1 (2011), 1039–1052.

[31] Bálint Tillman, Athina Markopoulou, Carter T Butts, and Minas Gjoka. 2017. Construction of directed 2K graphs. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1115–1124.

[32] David Bruce Wilson. 1996. Generating random spanning trees more quickly than the cover time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 296–303.

[33] Zhuojie Zhou, Nan Zhang, and Gautam Das. 2015. Leveraging history for faster sampling of online social networks. *arXiv preprint arXiv:1505.00079* (2015).